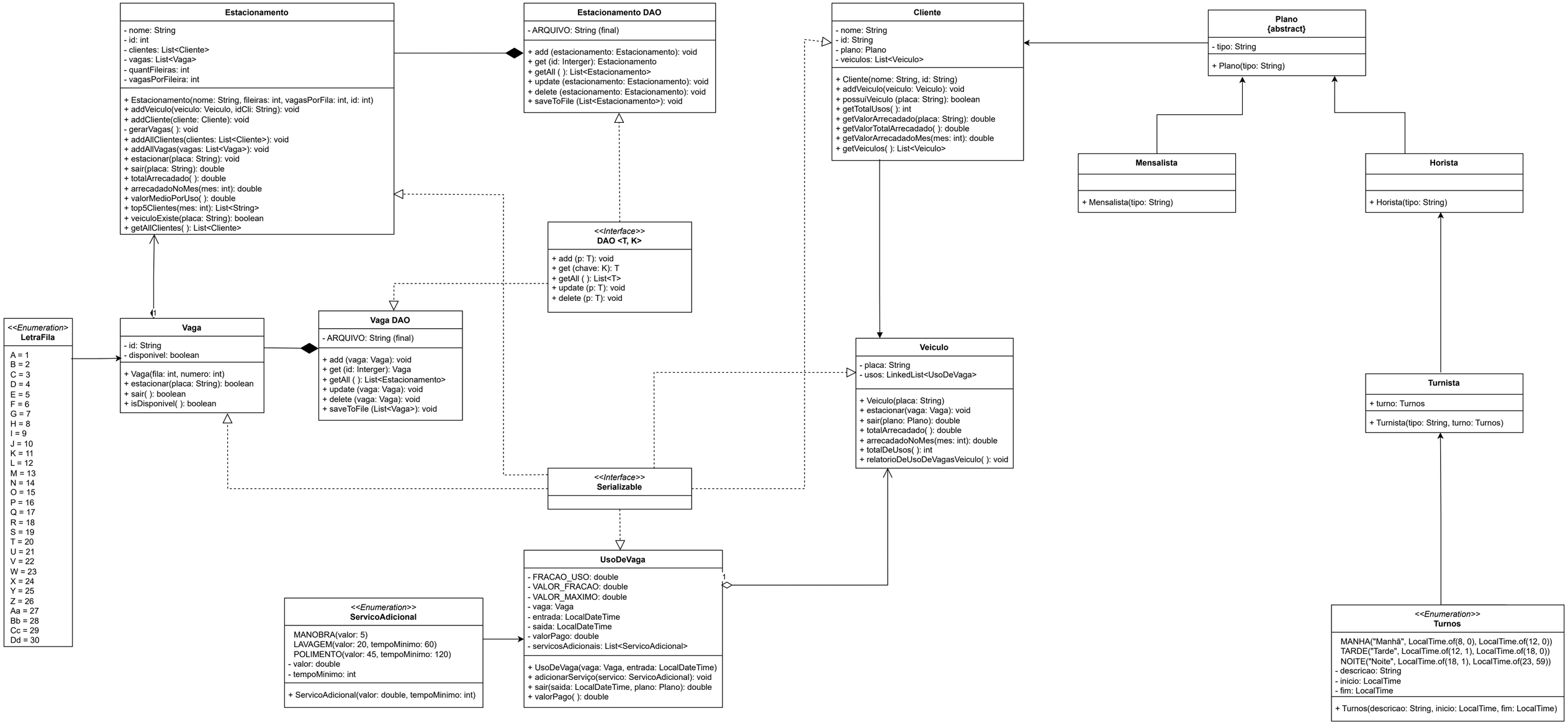
Estacionamento
- nome: String
- id: int
- clientes: List<Cliente>
- vagas: List<Vaga>
- quantFileiras: int
- vagasPorFileira: int
+ Estacionamento(nome: String, fileiras: int, vagasPorFila: int, id: int)
+ addVeiculo(veiculo: Veiculo, idCli: String): void
+ addCliente(cliente: Cliente): void
- gerarVagas( ): void
+ addAllClientes(clientes: List<Cliente>): void
+ addAllVagas(vagas: List<Vaga>): void
+ estacionar(placa: String): void
+ sair(placa: String): double
+ totalArrecadado( ): double
+ arrecadadoNoMes(mes: int): double
+ valorMedioPorUso( ): double
+ top5Clientes(mes: int): List<String>
+ veiculoExiste(placa: String): boolean
+ getAllClientes( ): List<Cliente>
Estacionamento DAO
- ARQUIVO: String (final)
+ add (estacionamento: Estacionamento): void
+ get (id: Interger): Estacionamento
+ getAll ( ): List<Estacionamento>
+ update (estacionamento: Estacionamento): void
+ delete (estacionamento: Estacionamento): void
+ saveToFile (List<Estacionamento>): void
Cliente
- nome: String
- id: String
- plano: Plano
- veiculos: List<Veiculo>
+ Cliente(nome: String, id: String)
+ addVeiculo(veiculo: Veiculo): void
+ possuiVeiculo (placa: String): boolean
+ getTotalUsos( ): int
+ getValorArrecadado(placa: String): double
+ getValorTotalArrecadado( ): double
+ getValorArrecadadoMes(mes: int): double
+ getVeiculos( ): List<Veiculo>
Plano
{abstract}
- tipo: String
+ Plano(tipo: String)
Mensalista
+ Mensalista(tipo: String)
Horista
+ Horista(tipo: String)
<<Interface>>
DAO <T, K>
+ add (p: T): void
+ get (chave: K): T
+ getAll ( ): List<T>
+ update (p: T): void
+ delete (p: T): void
1
<<Enumeration>
LetraFila
A = 1
B = 2
C = 3
D = 4
E = 5
F = 6
G = 7
H = 8
I = 9
J = 10
K = 11
L = 12
M = 13
N = 14
O = 15
P = 16
Q = 17
R = 18
S = 19
T = 20
U = 21
V = 22
W = 23
X = 24
Y = 25
Z = 26
Aa = 27
Bb = 28
Cc = 29
Dd = 30
Vaga
- id: String
- disponivel: boolean
+ Vaga(fila: int, numero: int)
+ estacionar(placa: String): boolean
+ sair( ): boolean
+ isDisponivel( ): boolean
Vaga DAO
- ARQUIVO: String (final)
+ add (vaga: Vaga): void
+ get (id: Interger): Vaga
+ getAll ( ): List<Estacionamento>
+ update (vaga: Vaga): void
+ delete (vaga: Vaga): void
+ saveToFile (List<Vaga>): void
Veiculo
- placa: String
- usos: LinkedList<UsoDeVaga>
+ Veiculo(placa: String)
+ estacionar(vaga: Vaga): void
+ sair(plano: Plano): double
+ totalArrecadado( ): double
+ arrecadadoNoMes(mes: int): double
+ totalDeUsos( ): int
+ relatorioDeUsoDeVagasVeiculo( ): void
Turnista
+ turno: Turnos
+ Turnista(tipo: String, turno: Turnos)
<<Interface>>
Serializable
UsoDeVaga
- FRACAO_USO: double
- VALOR_FRACAO: double
- VALOR_MAXIMO: double
- vaga: Vaga
- entrada: LocalDateTime
- saida: LocalDateTime
- valorPago: double
- servicosAdicionais: List<ServicoAdicional>
+ UsoDeVaga(vaga: Vaga, entrada: LocalDateTime)
+ adicionarServiço(servico: ServicoAdicional): void
+ sair(saida: LocalDateTime, plano: Plano): double
+ valorPago( ): double
1
<<Enumeration>>
ServicoAdicional
MANOBRA(valor: 5)
LAVAGEM(valor: 20, tempoMinimo: 60)
POLIMENTO(valor: 45, tempoMinimo: 120)
- valor: double
- tempoMinimo: int
+ ServicoAdicional(valor: double, tempoMinimo: int)
<<Enumeration>>
Turnos
MANHA("Manhã", LocalTime.of(8, 0), LocalTime.of(12, 0))
TARDE("Tarde", LocalTime.of(12, 1), LocalTime.of(18, 0))
NOITE("Noite", LocalTime.of(18, 1), LocalTime.of(23, 59))
- descricao: String
- inicio: LocalTime
- fim: LocalTime
+ Turnos(descricao: String, inicio: LocalTime, fim: LocalTime)